

# 자율 주차 시스템을 위한 주차 경로 추종 방법의 비교 연구

## A Comparative Study of Parking Path Following Methods for Autonomous Parking System

김민성<sup>1</sup>·임규범<sup>2</sup>·박재흥<sup>†</sup>

Minsung Kim<sup>1</sup>, Gyubeom Im<sup>2</sup>, Jaeheung Park<sup>†</sup>

**Abstract:** Over the last years, a number of different path following methods for the autonomous parking system have been proposed for tracking planned paths. However, it is difficult to find a study comparing path following methods for a short path length with large curvature such as a parking path. In this paper, we conduct a comparative study of the path following methods for perpendicular parking. By using Monte-Carlo simulation, we determine the optimal parameters of each controller and analyze the performance of the path following. In addition, we consider the path following error occurred at the switching point where forward and reverse paths are switched. To address this error, we conduct the comparative study of the path following methods with the one thousand switching points generated by the Monte-Carlo method. The performance of each controller is analyzed using the V-rep simulator. With the simulation results, this paper provides a deep discussion about the effectiveness and limitations of each algorithm.

**Keywords:** Path Following, Motion Control, Model Predictive Control, Intelligent Vehicle, Autonomous Parking System

### 1. 서 론

최근에 자율 주행 기술들이 활발히 개발되면서 고속도로뿐만 아니라 도심에서도 사용 가능한 자율 주행 기술들이 개발되고 있다<sup>[1]</sup>. 대표적인 도심 도로에서의 자율 주행 기술 연구로 주차 보조 시스템 연구가 있고, 자기 위치 인식 기술이나 주차 공간 인식 등과 같은 기술과 통합된 완전 자율 주차 시스템 연구로 확장되고 있다<sup>[2]</sup>. 자율 주차 시스템 연구는 특정 제한 조건에서만 최적으로 주차가 가능한 한계를 가진 기존의 상용 주차 보조 시스템과 다르게 어떠한 초기(Initial) 위치에서도 주차 경로를 생성하여 주차하는 것을 목표로 하고 있다.

자율 주차는 주차 공간 주변 환경을 인지(Recognition)하는

파트와 주행 가능한 주차 경로 생성(Path Planning) 및 경로 추종(Path Following) 파트로 구성된다. 인식된 주차 공간을 바탕으로 주행 가능한 주차 경로가 생성되면, 주변 차량 및 장애물과의 충돌을 피하면서 정확한 위치에 주차를 해야 하기 때문에 주차 경로는 정밀하게 추종 되어야 한다.

일반적인 전방 경로 추종 문제보다 상대적으로 주차 경로 추종 문제는 더 어렵다<sup>[3]</sup>. 그 이유로 다음과 같이 세 가지를 들 수 있다. 첫 번째, 주차 경로 추종은 경로 추종(Path Tracking/Following) 문제이면서 동시에 차량의 자세(Posture: 위치 및 방위)를 어떤 하나의 주차 목표 자세로 수렴시키는 점-안정화(Point Stabilization) 문제다. 이 문제는 Brockett's necessary condition<sup>[4]</sup>에 의해 제어 입력보다 자유도가 더 큰 논홀로노믹(Non-Holonomic) 시스템의 안정화 문제로서 시-불변(Time-invariant) 입력 피드백 제어 법칙에 의해서는 안정화 될 수 없다는 점에서 어려운 문제이다. 두 번째, 주차 환경에서 생성된 주차 경로는 일반 주행 경로보다 길이가 짧고 곡률이 크다. 만일, 주차를 수행하는 로봇이 차량형(Car-like) 로봇이라면, 일반적인 차륜형(Differential Drive) 모바일 로봇보다 큰 선회 반경의 한계를 가지고 있으며 주차 경로는 길이가 짧기 때문에

Received : Nov. 18. 2019; Revised : Jan. 30. 2020; Accepted : Feb. 3. 2020

1. PhD student, Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea (minsungkim@snu.ac.kr)

2. MS student, Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea (edwardim@snu.ac.kr)

† Professor, Corresponding author: Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea. Advanced Institutes of Convergence Science and Technology. (park73@snu.ac.kr)

로봇이 주차 경로 추종 오차를 빠르게 줄이지 못하면 주차가 실패될 수 있다. 이렇게 일반적인 자율 주행의 경로 추종 상황보다 상대적으로 경로 길이가 짧은 주차 경로 추종의 특성을 고려하기 위해서는 과도 응답 특성이 좋은 제어기가 필요하다<sup>5,6)</sup>. 마지막으로, 주차 경로 추종은 자율 주행 자동차의 적응형 순항 제어(Adaptive Cruise Control), 차선 유지 시스템(Lane Keeping System) 등과 같은 다양한 주행 기술 분야 중, 가장 정밀한 주행 제어 성능을 요구한다.

본 논문에서는 자율 주차에 적용될 수 있는 경로 추종 방법에 대해서 소개하고, 각 제어 방법별 정성적/정량적 분석을 수행하였다. 특히, 위에서 언급한 곡률이 크며 경로의 길이가 짧고 경로의 끝 점이 있는 후방 주차 경로에 대해서 경로 추종 방법들의 성능 비교를 수행하였다. 또한, 실제 차량으로 주차 경로를 추종하는 상황을 고려하여 제어 입력 외란에 대한 제어기의 성능도 비교 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 경로 추종 방법으로 제안되었던 알고리즘들의 특징에 대해 설명을 하고, 3장에서는 이러한 경로 추종 방법 중, 주차 경로 추종에 적용될 수 있는 알고리즘들에 관해서 기술한다. 4장에서는 로봇 시뮬레이터인 V-rep 환경에서 각 제어 방법에 사용되는 최적의 제어 파라미터를 찾고 이를 이용하여 제어 입력에 외란이 있을 시, 주차 경로 추종 성능을 비교한다. 그 다음, 주차 할 때 전/후진 전환 지점에서 임의의 초기 주차 경로 추종 오차에 대한 각 제어 방법별 성능 평가 및 장/단점 분석을 하고, 5장에서 본 논문을 결론짓는다.

## 2. 관련 연구

경로 추종 방법으로 대표적으로 알려진 Stanley<sup>7)</sup>, Pure-pursuit<sup>8)</sup>와 같은 geometric 방법과 adaptive PID 제어기 등은 구현이 쉬우며 계산 비용이 적다는 장점이 있다. 그러나, 이 방법들은 큰 곡률에서 불안정성을 보일 수 있으며 오차에 대해 응답을 하는 방법이기 때문에 목표 경로에 급격한 곡률이 있을 때, 과도 응답 특성이 좋지 않다. 특히, 2005년 미국 국방고등기획국이 후원하여 개최한 자율 주행 자동차 대회에서<sup>9)</sup>, Stanford 팀의 경로 추종 제어기로 알려진 Stanley 방법은 후방 주차 경로 추종 시, 제어 입력 출력 관계가 비-최소 위상 시스템(Non-Minimum Phase System)을 구성하기 때문에 후방 경로 추종에 적합하지 않다<sup>5)</sup>. Pure-pursuit는 휴리스틱(Heuristic)하게 정한 look-ahead 거리에 따라 경로 추종 성능이 크게 좌우되며 cutting corner와 같은 문제를 발생시킬 수 있다. 또한, 이 제어기는 look-ahead 거리를 이용함으로써 과도 응답(Transient Response) 특성을 좋게 하는 장점이 있으나, 근본적으로 주차 경로 추종 제어기로 적용하기에는 방위 오차를 고

려하지 않는 한계가 있다. 한편, Pure-pursuit와 유사하게 방법으로, look-ahead 거리 개념을 이용하면서 동시에 차량과 가장 가까운 목표 점과 특정 거리에 있는 목표 점에서의 오차까지 고려하여 만든 경로 추종 알고리즘(Input-output feedback linearization with preview, 이하 Preview)도 연구되었다<sup>10)</sup>. 이 방법은 예견 거리(Preview Distance)에 있는 목표점에서의 거리 오차를 고려하여 과도 응답 특성이 좋다는 장점이 있다. 또한, 차륜형 모바일 로봇의 경로 추종 방법으로 사용된 Kanayama<sup>11)</sup> 제어기는 목표 요(yaw)가 포함된 제어 방법으로 경로 추종 시, 목표 경로의 곡률을 이용한다는 점에서 과도 응답 특성이 좋은 제어 방법이다.

Fuzzy 제어기<sup>12)</sup>와 Sliding mode 제어기는 차량 모델에 기반하지 않는 방법으로써 좋은 제어 성능을 보였으나<sup>13)</sup>, 퍼지 제어 입력을 설정하기 위하여 매우 많은 실제 경험을 필요로 하며, Sliding mode 제어기는 채터링(Chattering) 문제로 인하여 승차감 저해가 수반되고, 실제 차에 적용하기에 한계가 있다.

이에 반해, 차량 모델에 기반한 주차 경로 추종 방법으로 대표적 방법인 Linear Quadratic Regulator (LQR)<sup>14)</sup> 방법 및 Model Predictive Control (MPC) 방법이 있다. LQR은 대수 리카티 방정식(Algebraic Riccati Equation)을 풀어 제어 이득을 구하는 방법으로 제어 오차가 발생 되었을 때에만, 해당 오차에 반응하여 제어 입력을 구하기 때문에 제어 오차를 발생시킬 수 있다. 이에 반해, MPC는 시스템의 실제 물리적인 제약 조건을 고려하면서 동시에 비용 함수를 최소화 하는 최적의 제어 입력을 구한다는 장점이 있으며 근래에 주차 경로 추종에 적용 되었고 제어 성능이 확인 되었다<sup>11)</sup>.

기존의 경로 추종 방법에 대한 연구 외에 다양한 경로 추종 방법을 비교한 연구들도 있었다. Brezak et al.는 세 가지 제어기를 비교하여 가장 적은 경로 추종 오차를 보이면서 동시에 외란에 강한 제어기에 대한 성능 평가를 하였고<sup>15)</sup>, non-linear 제어 방법의 성능이 가장 낫다는 결론을 지었다. Heb et al.과 Calzolari et al.에서도 차량 모델의 불확실성에 대한 제어기의 성능을 비교하였고, 제어기별 성능을 분석 할 수 있는 벤치마크 테스트 프레임워크를 제안했으나, 주차 경로와 같이 경로가 끝나는 목표 경로에 대한 분석이 이뤄지지 않았으며 벤치마크 테스트를 할 수 있는 해당 사이트는 더 이상 이용할 수 없다<sup>16,17)</sup>. Dominguez et al.는 경로 추종 알고리즘으로 Stanley Pure-pursuit, Slide mode control, Lateral speed control 방법에 대해서 시뮬레이션과 실제 차량 테스트로 전방 경로 추종 성능을 비교하였다<sup>18)</sup>.

기존의 경로 추종 방법들은 대부분 전방 경로 추종에서만 제어 성능을 비교했기 때문에 공통적으로 경로의 곡률이 크고, 경로 길이가 짧은 주차 경로 추종에 바로 적용하기에는 한계가 있다. 다음 장에서는 앞에서 언급한 것처럼 주차 경로 추

중 제어기로써의 가져야 할 특성이 고려된 세 가지 제어 방법 (Kanayama, Preview 및 MPC)들을 소개하며 알고리즘별 장단 점에 대해 논하고자 한다.

### 3. 주차 경로 추종 방법

#### 3.1 Kanayama Controller

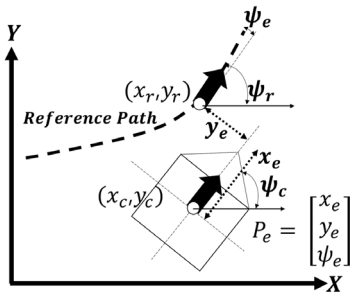
Kanayama 제어기는 1990년 Kanayama et. al. 가 제안 하였 다<sup>[11]</sup>. 이 경로 추종 제어기는 geometric 경로 추종 방법으로써, 구현이 쉽고 다른 제어기에 비해 상대적으로 계산 비용이 적다 는 장점이 있다. [Fig. 1]에서 보면, 로봇은 현재 자세( $x_c, y_c, \psi_c$ ) 에서 세 가지 자유도(Degree of Freedom)를 지니고 있고  $x_r, y_r$  및  $\psi_r$ 은 기준 자세다. 제어 입력은 다음과 같이 계산된다. 우선, 식 (1)에서 보면, 관성 좌표계에 대한 기준 궤적과 현재 모 바일 로봇의 위치 및 자세의 차이는 모바일 로봇 기준 좌표계 대 해 표현될 수 있다. 모바일 로봇의 위치 및 방위 오차는 모바일 로봇의 기준 좌표계에 대해서 정의된다( $x_e, y_e, \psi_e$ ). 이렇게 정의된 오차 자세( $P_e$ )를 시간에 대해 미분하고, 이동 로봇의 논홀노믹 구속 조건( $\dot{x}_c \sin \theta_c - \dot{y}_c \cos \theta_c = 0$ )을 대입한 뒤, 리 아푸노프(Lyapunov) 함수를 구성하여 정리하면 식 (2)가 얻어 지며 자세한 수식 유도 과정은 해당 논문을 참고한다<sup>[11]</sup>.

$$P_e = \begin{bmatrix} x_e \\ y_e \\ \psi_e \end{bmatrix} = \begin{bmatrix} \cos(\psi_c) & \sin(\psi_c) & 0 \\ -\sin(\psi_c) & \cos(\psi_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} x_r \\ y_r \\ \psi_r \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ \psi_c \end{bmatrix} \right) \quad (1)$$

$$= \begin{bmatrix} \cos(\psi_c)(x_r - x_c) + \sin(\psi_c)(y_r - y_c) \\ -\sin(\psi_c)(x_r - x_c) + \cos(\psi_c)(y_r - y_c) \\ \psi_r - \psi_c \end{bmatrix}$$

$$\begin{aligned} v_c &= v_r \cos(\psi_e) + K_x x_e, \\ \omega_c &= \omega_r + v_r (K_y y_e + K_\psi \sin(\psi_e)) \end{aligned} \quad (2)$$

Kanayama et al. 논문에서  $v_c$ 와  $\omega_c$ 는 각각 선 속도와 각속도 제어 입력이며  $v_r$ 와  $\omega_r$ 는 목표 선 속도와 목표 각속도를 의미



[Fig. 1] Kanayama error posture: error posture with respect to the Kanayama controller

한다.  $K_x, K_y, K_\psi$ 는 양수 상수인 제어 파라미터다. 이 제어 파 라미터를 사용하여 Kanayama 제어기는 일정한 곡률을 가진 경로 추종에 대한 안정성을 보장 할 수 있다<sup>[11]</sup>. Kanayama 제 어기는 모바일 로봇의 현재 선속도와 각속도가 요구되는 속 도와 완벽히 일치한다는 가정하에 제안되었으며 전통적으 로 차륜형 모바일 로봇의 궤적 추종 제어에 사용되어 왔다. Kanayama 제어기를 차량과 같은 로봇에 적용하기 위해서 위 에서 언급한 제어 입력 중 각속도는 바로 제어 입력으로 적용 되지 않으며 해당 각속도를 생성하기 위한 조향각( $\delta$ )이 제어 입력으로 사용 되어야 한다. 여기에서 사용되는 모델은 3.3절 에서 설명 되어 있다. 이러한 차륜형 모바일 로봇과 차량과 같 은 형태의 로봇의 기구학적 유사성으로 인하여 Kanayama 제 어기는 식 (3)과 같은 관계를 가질 수 있다. 최종 제어 입력  $\delta$ 는 다음과 같이 구해진다.

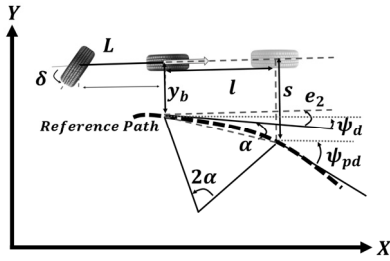
$$\omega_c = \kappa v = \frac{\tan \delta}{L} v, \quad \delta = \tan^{-1} \frac{\omega_c L}{v} \quad (3)$$

$\kappa$ 는 순간선회 반경(Instantaneous Radius of Turn)의 역수로 곡률(Curvature)을 의미한다.  $L$ 은 차량의 축간 길이(Wheel base)를 의미한다.

앞서 언급된 주차 경로 추종용 제어기의 특징으로써 Kanayama 제어기에서는 목표 yaw rate ( $\omega_r$ ) 부분이 있다. 식 (2)에서 보 면,  $\omega_r$  부분은 횡 방향의 거리 오차와 방위 오차를 고려한 부 분이 더해져 있다. 이러한 형태가 경로 추종 중 가장 가까운 목표점을 추종하다가 발생한 오차를 줄여주는 feedback 제어 기 역할을 하고, 이미 주어진 목표 경로를 고려한 feed-forward 형태의 역할을 하는 제어기로 구성된다. 즉, 경로 추종 오차가 없을 때에는 경로와 매우 가까운 곳에 차량이 놓인 상황이며 이때, 목표 경로로부터 곡률을 기반으로 한 조향각을 계산할 수 있다. 결과적으로 feed-forward 부분은 주차 경로 추종 시, 과도 응답 특성을 좋게 만드는 역할을 한다.

#### 3.2 Input-output feedback linearization with preview controller (Preview controller)

Preview 제어기는 후방 경로 추종을 하기 위해 Rajamani et al.에 의해 개발되었다<sup>[10]</sup>. 이 제어기는 차량으로 곡률이 있는 경로를 추종 할 때, 차량 뒷바퀴 중심에서 일정한 예견 거리에 있는 목표점을 설정하고 이 목표점에서 정의한 횡 방향 거리 오차를 줄인다는 점이 특징이다. 우선, 차량 뒷바퀴에서 예견 거리( $l$ )만큼 떨어진 점에 가상의 바퀴가 있다고 가정하고, 가 상의 바퀴에서 목표 경로에 수직으로 내린 목표 경로 점 (Waypoint)을 예견 위치(Preview Point)라고 정의한다[Fig. 2].



[Fig. 2] Geometry relationship of the preview controller

가상의 바퀴에서부터 예견 위치까지의 거리를 횡 방향 거리 오차  $s$ 라고 정의한다.  $y_b$ 는 차량 뒷바퀴에서 목표 경로와 가장 가까운 점이고, 이 경로 점의 방위 정보가 목표 방위  $\psi_d$ 이며,  $s$ 를 이용해서 구한 예견 위치에서의 목표 방위는  $\psi_{pd}$ 이다.  $s$ 는 배각 공식과  $\alpha = \frac{\psi_d - \psi_{pd}}{2}$  관계 및 small angle theorem에 의해 다음과 같이 식 (4)로 계산된다.

$$s = y_d + l(e_2 + \frac{\psi_d - \psi_{pd}}{2}) \quad (4)$$

여기서  $e_2$ 는 뒷바퀴에서 목표 경로와 가장 가까운 점과의 방위 오차이다.  $\dot{s} = -\lambda s$ 로 놓으면 다음 식 (5)와 같은 관계를 얻을 수 있다.  $\lambda$ 는 양수인 제어 파라미터이다.

$$\dot{y}_b + l \left( \dot{e}_2 + \frac{\dot{\psi}_d - \dot{\psi}_{pd}}{2} \right) = -\lambda s \quad (5)$$

$e_2 = \psi - \psi_d$ 로 놓고 이를 미분한 뒤,  $\dot{\psi} = -\frac{v}{L} \tan \delta$  관계를 고려하여 식 (5)에 넣으면 제어 입력인 식 (6)으로 정리할 수 있으며  $v$ 는 차량의 속력이고  $L$ 은 차량의 축간 길이이다.

$$\delta = -\frac{L}{lv} \tan^{-1} \left( -\lambda y_b - \lambda e_2 - \lambda \frac{\psi_d - \psi_{pd}}{2} - \dot{y}_b + \frac{(\dot{\psi}_d + \dot{\psi}_{pd})l}{2} \right) \quad (6)$$

제어기의 자세한 유도는 해당 논문을 참고한다<sup>[10]</sup>. Preview 제어기는 주차 경로 추종 제어기로서 예견 거리에 위치해 있는 목표점에 대한 오차를 정의하고 이를 줄이는 제어 입력을 구하기 때문에 과도 응답 특성이 개선된다<sup>[5]</sup>. 이 제어기는 임의의 예견 거리를 사용한다는 점에서 대표적인 geometric 제어기인 Pure-pursuit 방법과 유사하다. 그러나, Preview 제어기는 예견 거리에 있는 점에서의 방위 오차로 생긴 횡 방향 거리 오차를 줄이기 때문에 방위 오차는 상관없이 목표점에 선회 운동으로 도달할 수 있는 조향각을 구하는 Pure-pursuit보다 더 나은 제어 성능을 기대할 수 있다.

### 3.3 Model Predictive Controller

MPC를 이용하여 경로 추종 할 때, 주차 공간에서의 차량은 저속으로 이동하기 때문에 차량 기구학 모델을 이용하여 차량의 움직임을 표현될 수 있다. 본 연구에서는 효율적인 계산을 위해서 선형 MPC를 구현 하였다. MPC에 사용된 모델은 비선형 차량 기구학 모델이다[Fig. 3]. [Fig. 3]은 차량의 전/후 두 바퀴가 각각 하나의 바퀴로 표현되는 모델이며  $x_c, y_c$ 는 각각 차량의 뒷바퀴 중심의 위치를 의미한다.  $x_{r(i)}, y_{r(i)}$ 에서  $r$ 은 목표 위치 점을 의미하며,  $r(i)$ 는  $i$ 번째 목표 위치 점을 의미한다. 기구학 모델은 식 (7)로 정리될 수 있다.

$$\dot{x} = v \cdot \cos(\psi), \dot{y} = v \cdot \sin(\psi), \dot{\psi} = v \cdot \frac{\tan(\delta)}{L} \quad (7)$$

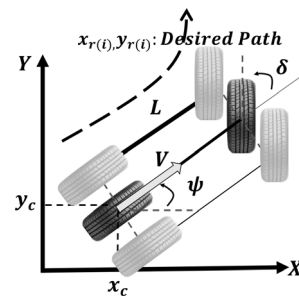
위와 같은 비선형 모델은 Talyor 확장을 이용한 선형화 (Linearization)를 통해 선형 MPC에 적용될 수 있다. Talyor 확장은 비선형 차량 모델과 평형 상태(Equilibrium state)를 의미하는 목표 경로 점과 관계를 통해 정의되는데, 이는 차량 모델과 선형화하고자 하는 목표 모델( $X_r = [x_r, y_r, \psi_r]^T$ ,  $U_r = [v_r, \delta_r]^T$ )과의 차이 형태로 정의되며  $\tilde{x} = X - X_r$ 과  $\tilde{u} = U - U_r$ 로 표현될 수 있다( $X = [x, y, \psi]^T$ ,  $U = [v, \delta]^T$ ). 이와 같이 선형화된 식을 이산화(Discretization)하면 다음과 같은 식으로 나타낼 수 있다.

$$\tilde{x}(k+1) = A(k)\tilde{x}(k) + B(k)\tilde{u}(k) \quad (8)$$

식 (8)은 선형 시변(Linear Time-Varying) 시스템을 의미하고,  $A(k)$ 와  $B(k)$  행렬은 다음과 같다.

$$A(k) = \begin{bmatrix} 1 & 0 & -v_{r,k} \sin(\psi_{r,k}) T \\ 0 & 1 & v_{r,k} \cos(\psi_{r,k}) T \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$B(k) = \begin{bmatrix} \cos(\psi_{r,k}) T & 0 \\ \sin(\psi_{r,k}) T & 0 \\ \frac{\tan(\delta_{r,k}) T}{L} & \frac{v_{r,k} (1 + \tan^2(\delta_{r,k})) T}{L} \end{bmatrix}$$



[Fig. 3] Kinematic bicycle model for the MPC

$L$ 은 차량의 축간 길이를 의미한다.  $v_{r,k}$ 과  $\psi_{r,k}$ 는 각각  $k$  순간의 목표 속력과 목표 경로 위치의 목표 방위 값이며  $T$ 는 샘플링 주기,  $\delta_{r,k}$ 는 목표 조향각이다. 여기서  $\delta_{r,k}$ 는 차량이 목표 경로점 위에 놓였을 때, 다음 목표점으로 경로 추종하는데 필요한 조향각을 의미한다.

주차 경로 추종은 2장에서 언급된 바와 같이 일반적인 고속도로 경로 추종에 비해 과도 응답 특성이 더 요구되기 때문에 경로의 예견 정보가 필요하다. MPC 방법은 차량 모델을 이용하여 차량의 미래 거동을 예측하고 미리 정의해 놓은 비용 함수를 최소화하는 최적 제어 입력을 구하는 방식이기 때문에 경로 추종 시, 경로의 예견 정보가 이용됨을 알 수 있다. 비용 함수는 다음과 같다.

$$\begin{aligned} \min_{\tilde{u}(k)} J(k) &= \sum_{i=1}^{H_p-1} \tilde{x}(k+i)^T Q \tilde{x}(k+i) \\ &+ \tilde{x}(k+H_p)^T P \tilde{x}(k+H_p) \\ &+ \sum_{i=0}^{H_c-1} \tilde{u}(k+i)^T R \tilde{u}(k+i) \end{aligned} \quad (10)$$

$$\begin{aligned} \text{s.t. } \tilde{x}_0 &= \tilde{x}(k), \\ \tilde{x}(k+i+1) &= A(k+i)\tilde{x}(k+i) + B(k+i)\tilde{u}(k+i) \\ \tilde{u}_{\min} &\leq \tilde{u}(k+i) \leq \tilde{u}_{\max} \end{aligned}$$

위에서  $\tilde{x}_0$ 는 현재  $k$ 에서의 시스템 상태 값이다.  $i$ 와  $j$ 의 범위는 각각  $i \in [0, H_p - 1]$ ,  $j \in [0, H_c - 1]$ 이며  $H_p$ 는 예측 구간,  $H_c$ 는 제어 구간이고 동일하게 설정하였다.  $Q$ 는 시스템 상태의 가중 매트릭스이고  $P$ 는 종단(Terminal) 시스템 상태에 대한 가중 매트릭스이며  $Q$ 와  $P$ 는 동일하게 설정하였다.  $R$ 은 제어 입력 상태 가중 매트릭스이다. 입력 제약조건인  $\tilde{u}_{\min}$ ,  $\tilde{u}_{\max}$ 에는 최대 조향각( $\pm 540^\circ$ )을 이용하였다. MPC는 위의 비용 함수를 최소화하는 제어 입력 벡터를 구하고 이 중 첫 번째 값을 최종 제어 입력으로 이용한다. 이를 위해 식(10)은 QP(Quadratic Programming)에 대입하기 위한 식으로 변환한 뒤 이를 최소화하는 제어 입력을 구한다. 본 연구에서는 MPC의 최적화 계산을 위한 solver로써 CVXGEN의 QP를 이용하였다<sup>19)</sup>. 제어 주기는 10Hz로 적용하였다. 최종 제어 입력은 최적화를 통해 얻은 최적 제어 입력 벡터인  $\tilde{u}_k^*$  ( $k \in [0, H_c - 1]$ )의 첫 번째 조향각과 앞에서 구한 목표 조향각( $\delta_{r,k}$ ) 값의 첫 번째 값을 더해서 구한다.

이러한 MPC 기반 경로 추종 방법은 차량의 모델을 이용하여 차량의 미래 거동과 목표 경로와의 오차를 최소화하면서 동시에, 조향각 범위나 조향 각속도 범위와 같은 물리적인 제약(Constraints)을 고려할 수 있다는 장점이 있다. 그러나, 본 연구에서는 효율적인 제어 입력 계산을 위해 선형 MPC를 수

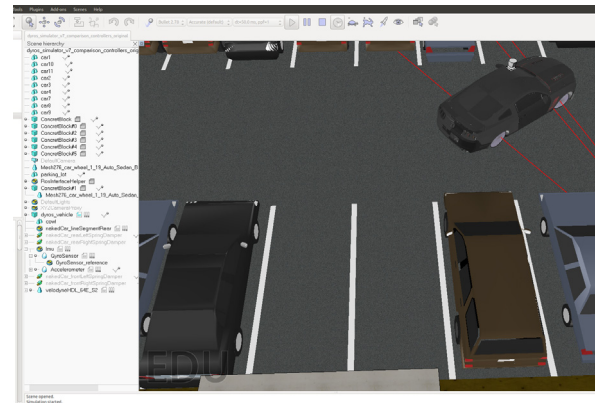
행하였고, 이를 위해 목표점에 대해서 선형화를 수행한 모델을 사용한다는 점에서 목표점과 멀리 떨어져 있는 상태에서는 선형화 모델의 기본 가정에 위배될 수 있는 한계가 있다. 다음 장에서는 세 가지 주차 경로 추종 방법의 성능 비교를 수행하였다.

## 4. 실험 및 결과

### 4.1 실험 환경 및 제어기 파라미터 결정 방법

본 연구는 세 가지 경로 추종 알고리즘을 이용한 경로 추종 성능을 테스트하기 위해서 ROS (Robot Operating System) 기반의 V-rep 시뮬레이터를 사용하였다<sup>20)</sup>. 차량 모델은 V-rep에서 제공하는 Simple Ackermann Steering 모델을 사용하였고, 기구학 정보는 현대자동차 그랜저 HG240의 제원을 이용하였다. 시뮬레이션을 수행한 컴퓨터 사양은 Intel i7 4.2GHz, 16 GB RAM이다. 시뮬레이션의 주차장은 경기도 수원시 영통구에 위치한 차세대융합기술연구원 주차장 환경을 실제 측정하여 제작되었다[Fig. 4].

한편, 주차 경로는 대표적인 샘플링 기반 경로 생성 알고리즘인 RRT (Rapidly-exploring Random Trees) star를 사용하였고<sup>21)</sup>, 차량의 구속 조건을 고려하였으며 경로 계획 라이브러리인 Open Motion Planning Library (OMPL)를 이용하여 구현되었다<sup>22)</sup>. 주차 시, 차량의 거동은 저속이기 때문에 Ackermann 조향으로 차량의 순간 선회 반경을 계산할 수 있다. RRT star로 생성된 경로의 최대 곡률은 0.07이며, 이는 Ackermann 조향에 따라, 축간 거리 2.978 m인 차량에서 최대 조향이 30°일 때 최대 곡률이 0.2077(1/선회 반경) 이기 때문에 차량이 추종할 수 있는 경로이다. 경로는 여러 개의 경로 점들로 구성되어 있으며 실제 주차 환경에서는 매 순간 목표 경로 대비 자차의 상대 위치를 정확하게 알기 어렵지만, 시뮬레이션에서는 이를 정확하게 알고 있다고 가정하였다.



[Fig. 4] V-rep simulator environment

제어기는 일반적으로 제어 입력 계산에 관여하는 파라미터에 따라 제어 성능이 바뀐다. 본 절에서는 미리 정의된 비용 함수를 최소화하는 제어 이득을 구하고자 하며, 이를 위해 몬테카를로 시뮬레이션(Monte Carlo Simulation)을 이용하였다<sup>[23]</sup>. 몬테카를로 시뮬레이션은 무작위로 추출된 난수(Random Number)를 이용하여 원하는 파라미터의 값을 확률적으로 구하는 알고리즘 및 시뮬레이션 방법을 의미한다. 특히, 이 방법은 주어진 문제의 해석적인 해(Analytical Solution)를 구할 수 없거나 풀이가 복잡한 경우, 해를 근사적으로 계산할 때 사용된다. 본 연구에서는 몬테카를로 시뮬레이션을 통해 최적으로 계산하고자 하는 비용을 다음과 같이 정의하였다.

$$J_e = \sum_{k=0}^{w_f} \sqrt{z1 \cdot ((x_k - x_{ref,k})^2 + (y_k - y_{ref,k})^2) + z2 \cdot (\psi_k - \psi_{ref,k})^2} \quad (11)$$

$$J_c = \sum_{k=0}^{w_f} |\nabla \delta_k|$$

$$J_{total} = \alpha \cdot J_e + \beta \cdot J_c$$

식(11)에서 경로 추종 오차 비용( $J_e$ )은 각각의 제어기를 이용하여 한 번의 경로 추종 시행(Trial)에서의 누적된 경로 추종 비용을 나타내며 거리 오차 비용과 방위 오차 비용으로 구성된다.  $x_k, y_k$  및  $\psi_k$ 는 차량의 현재 위치와 방위이며,  $ref, k$  점은 매 제어 순간( $k$ )의 목표 경로의 정보를 의미한다.  $z1$ 와  $z2$ 은 오차 비용에 대한 가중치이고, 실험적으로 결정된 값으로 가중치에 따라서 추종 오차 비용이 달라질 수 있으며 본 실험에서는 1로 가정하였다( $z1 = 1, z2 = 1$ ).  $w_f$ 는 목표 경로 점 개수이다.  $\nabla \delta_k$ 는 제어 입력의 변화 값이고  $J_c$ 는 제어 입력의 변화 비용을 의미하며 일반적으로 제어 노력(Control Effort)이라고도 불린다.  $J_{total}$  비용은  $J_e$ 와  $J_c$ 에 적절한 가중치를 곱해 더하였다( $\alpha = 1, \beta = 0.1$ ). 몬테카를로 시뮬레이션은  $J_{total}$ 를 최소로 하는 제어 입력을 구하기 위해 각 제어기별 제어 이득 관련 파라미터에 난수를 입력한 경로 추종 시뮬레이션으로써 각각 1천 회씩 수행되었다.

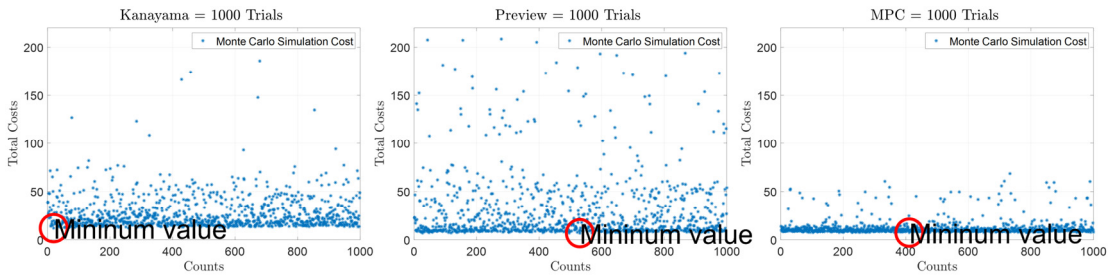
몬테카를로 시뮬레이션은 각 제어기가 최소(Minimum value, 빨간 원) 비용을 가질 때의 각 제어 입력 파라미터를 결정하였

다[Fig. 5]. 주차 경로는 전방 경로와 후방 경로로 구성되어 있으며, 경로 추종 비용( $J_{total}$ )을 계산하는 구간은 후방 경로 추종 구간에 국한하였다. [Fig. 5]에서 가로축은 시행 횟수이다. 세로축은 식(11)에서 정의한 비용  $J_{total}$ 를 의미하며 파란 점 하나가 한번의 시행을 의미한다. 본 연구에서 사용된 비용은 절대적이지 않고, 비용을 정의하기에 따라 각 제어기의 비용이 달라지며, 이에 대한 각 제어기별 제어 파라미터 값도 변할 수 있다.

Kanayama 제어기에는 식(2)에서  $K_y$ 와  $K_\theta$  두 파라미터에 각각 0.1 ~ 10.0 범위로 uniform 분포의 난수를 발생시켰다. Uniform 분포를 이용한 이유는 최적의 제어 파라미터는 특정 구간에 동일한 확률로 존재함을 가정했기 때문이다. 몬테카를로 시뮬레이션을 통해서 나온 Kanayama 제어기의 최적 파라미터( $K_y, K_\theta$ )는 6.993, 5.099이다.

Preview 제어기에서는 식(6)에서 확인할 수 있듯이, 예견 거리  $l$ 과 오차를 줄이는 수렴 속도 이득 파라미터인  $\lambda$ 에 대해서 각각 uniform 분포의 난수를 발생시켰다(0.1~ 3.0, 5~15). Preview 제어기의  $l$ 은 예견 거리를 의미하기 때문에 0보다 커야 한다. 주차 경로는 길이가 짧기 때문에, 주차 경로 추종 시뮬레이션을 통해 실험적으로 최적 파라미터의 분포 범위를 정하였다. Preview 제어기의 최적 파라미터( $l, \lambda$ )는 0.528, 6.31이다.

MPC 제어기에 대해서는 경로 추종 가중 파라미터인 Q행렬(3x3 행렬, 양의 정-부호 행렬)에서 3개의 파라미터 값과 제어 입력 가중 파라미터인 R행렬(2x2 행렬, 양의 정-부호 행렬)에 대한 1개의 파라미터, 총 4개의 파라미터에 uniform 분포의 난수를 발생시켰다(Q0:1~100, Q1:1~100, Q2:1~50, R1:0.001~0.1). Q0, Q1, Q2는 Q행렬의 대각선 성분이다. R0과 R1은 각각 R의 대각 성분 값이며, 속력 제어 입력에 관여하는 값인 R0은 1로 설정하였다. 조향 각에 대한 가중치 파라미터인 R1은 충분한 제어 입력으로 오차를 빨리 줄이기 위해 값 범위를 0.1 이하로 정하였다. 몬테카를로 시뮬레이션을 통해 얻은 MPC 제어기의 최적 파라미터(Q0, Q1, Q2, R1)는 65.640, 60.916, 22.659, 0.027이다. 세 제어기는 정-속력(3km/h) 주행을 가정하였고, 전방 주행을 하다가 전/후진 전환 지점에서는 멈추고 후방 주행을 하였다.



[Fig. 5] Monte Carlo simulation of path following methods (Left: Kanayama, Mid: Preview, Right: MPC)

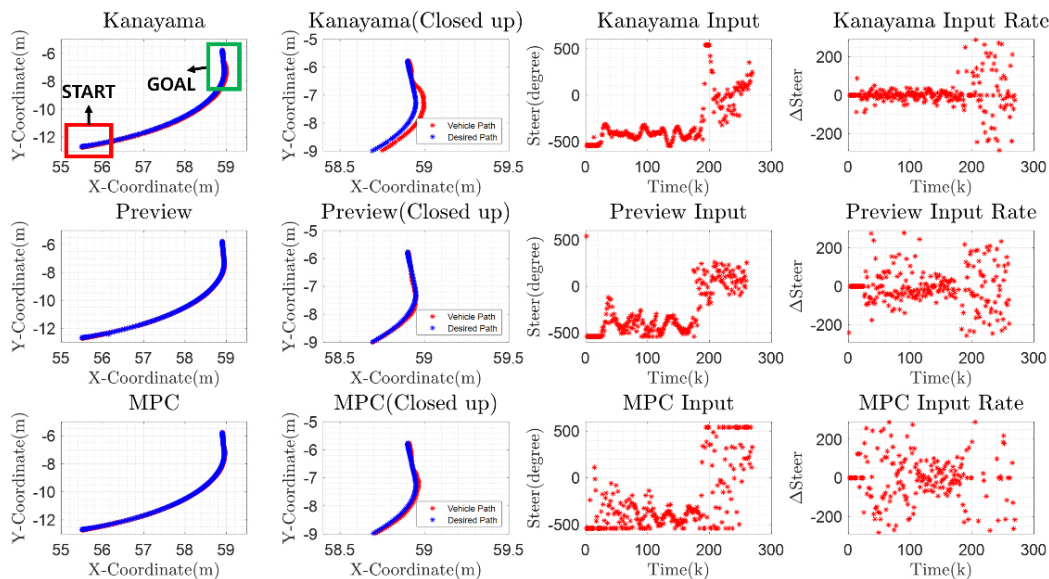
몬테카를로 시뮬레이션 결과를 통해 얻은 최적의 파라미터 값을 이용해서 주차 경로 추종을 수행한 결과는 [Fig. 6]에서 확인할 수 있다. [Fig. 6]의 첫 번째 행부터 세 번째 행까지 순서대로 각각 Kanayama, Preview, MPC 제어기의 경로 추종 결과를 의미한다. [Fig. 6]의 첫 번째 열은 세 가지 주차 경로 추종 방법 별 경로 추종을 수행한 궤적을 나타내며, 두 번째 열은 각 경로 추종을 수행 할 때, 주차 목표점 근처에서 추종된 경로를 확대한 것이다. 일반적인 자율 주행 경로 추종과 달리, 주차 경로 추종은 주차 목표점에서의 최종 오차가 주차 성공 여부에 중요한 역할을 하기 때문에 주차 목표점에서의 오차를 관찰할 필요가 있다. 세 번째 열은 각 제어기의 제어 입력을 의미하고 네 번째 열은 제어 입력 변화율이다.

세 가지의 제어 방법을 이용한 주차 경로 추종 결과는 [Table 1]에 정리되었다. [Table 1]에서 보면, Preview와 MPC 제어기가 평균 거리 오차 값이 1 cm 내외로 확인되었으며, 평균 방위 오차는 MPC가 가장 적었다. 또한, 주차 목표 지점에서의 최종 거리 오차는 Preview가 가장 적었으며, MPC가 주차 목표 지점에서의 가장 적은 최종 방위 오차를 보였다. 그러나, [Fig. 6]의 네 번째 열에서 확인할 수 있듯이, MPC 방법의 제어

입력 변화율이 세 가지 제어 방법 중 가장 넓은 분포를 보이고 있고 제어 입력 비용이 가장 크다(Kanayama: 7.87, Preview: 7.93, MPC: 8.69). 이는 그만큼 차량 핸들 각도에 큰 변화가 있음을 의미한다. 이런 부분은 MPC 제어 방법을 설계할 때, 제어 입력의 변화율을 고려한 제약 조건을 넣음으로써 개선할 수 있는 부분이다.

#### 4.2 초기 오차에 대한 세 가지 경로 추종 알고리즘 성능 비교 분석 실험

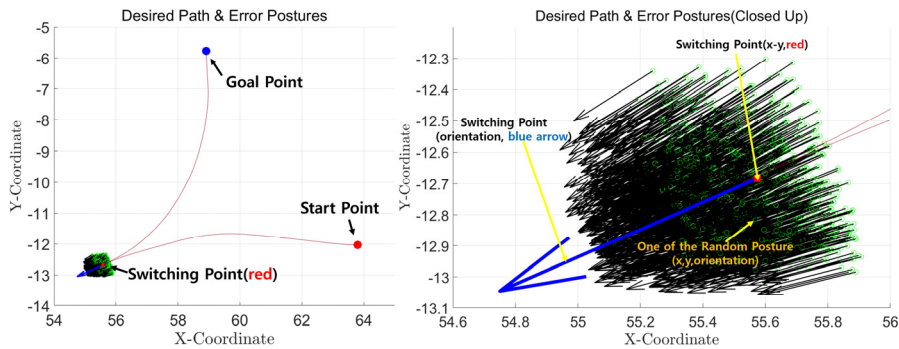
본 절에서는 4.1장에서 수행한 각 제어기별 주차 경로 추종을 위한 최적의 제어 파라미터를 이용하여 경로 추종 실험을 수행하였으며 초기 경로 추종 오차가 있는 상황을 가정하였다. 이렇게 목표 경로에서 떨어진 지점에서 경로 추종을 시작하는 것은 제어기의 추종 성능을 확인하는 하나의 방법이다. 초기 경로 추종 오차에 대한 성능은 1,000가지의 오차 자세 (Error Posture)에 대한 경로 추종의 시뮬레이션을 수행하여, 각 제어기 마다 성능의 변화를 통해 확인되었다. 여기서 오차 자세란, 주차 경로의 전/후진이 교차되는 하나의 점(목표 x,y,



[Fig. 6] Parking path following results by using the optimum control inputs which are acquired by the Monte Carlo simulation (Top row: Kanayama, Middle row: Preview, Bottom row: MPC)

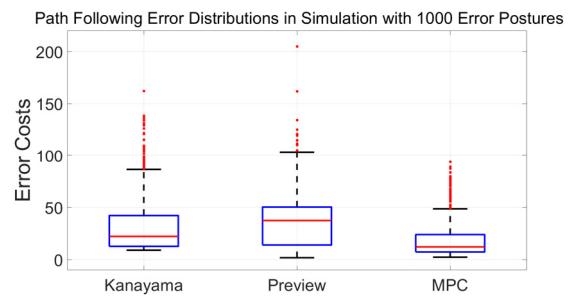
[Table 1] Parking path following results by using three methods with the optimum control parameters which are calculated by the Monte Carlo simulation

	Max. Distance Error (m)	Max. Orientation Error (°)	Avg. Distance Error (m)	Avg. Orientation Error (°)	Final Distance Error (m)	Final Orientation Error (°)
Kanayama	0.058	4.606	0.033	1.406	0.004	1.023
Preview	0.033	3.016	0.009	1.178	0.002	0.243
MPC	0.036	3.073	0.013	0.975	0.007	0.078



[Fig. 7] Parking path and the parking scenario (left) and 1,000 error postures trials (right)

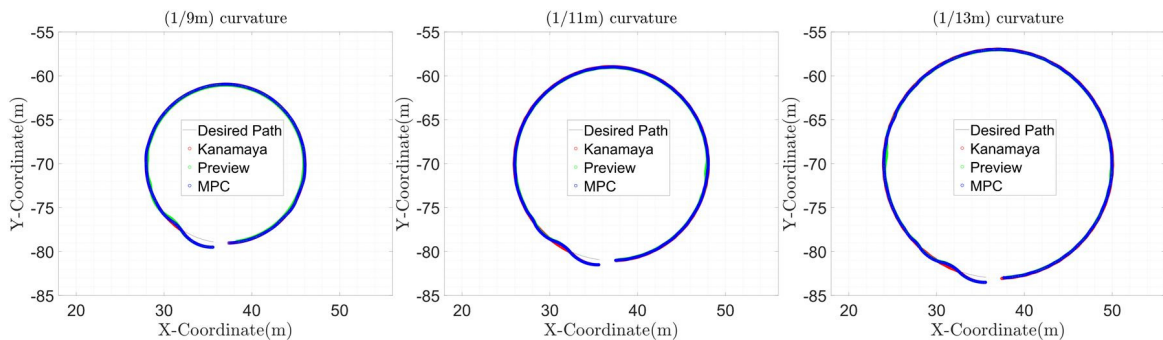
및 orientation)을 기준으로 일정한 범위( $\pm 0.3$  m의 거리 오차,  $\pm 10^\circ$ 의 방위 오차) 값을 더해서 임의로 만든 자세를 의미하며 차량은 이러한 1,000가지 초기 오차를 가정한 자세에서 후방 경로 추종을 시작하게 된다. [Fig. 7]의 왼쪽 그림은 4.1절에서 수행한 몬테카를로 시뮬레이션에서의 전방 및 후방 목표 경로이며 빨간 점에서 차량이 전방 주차 경로 추종을 시작하고 ‘Goal Point’가 주차 목표점이다. 오차 자세에서의 시뮬레이션은 ‘Switching point’라고 표기된 부분 주변에서 시작된다. [Fig. 7] 오른쪽 그림에서의 빨간 점은 차량의 전/후진 전환 지점에서의 목표 위치를 의미하며, 파란 화살표는 그 순간의 목표 방위를 의미한다. 초록 점들은 빨간 점 중심으로 임의로 생성한 1,000가지의 오차 위치를 의미하며, 검은 화살표는 이에 대한 각각 차량의 시작 방위를 의미한다. [Fig. 8]은 경로 추종 제어기별 초기 오차 자세에서의 경로 추종 오차 비용(=거리 오차+방위 오차)에 대한 box 그래프이며 Preview 제어기의 최소 비용 값이 가장 적었다(Kanayama: 9.464, Preview: 2.340, MPC: 3.154). 그러나, 경로 추종 오차 비용에서 75% 위치의 비용 값(Kanayama: 42.252, Preview: 50.374, MPC: 24.018) 및 최대 비용 값(Kanayama: 86.785, Preview: 103.544, MPC: 49.048)은 세 제어기 중 Preview가 가장 높다. 4.1절의 몬테카를로 시뮬레이션을 통해 얻은 최적 파라미터 기반으로 목표 경로 점과 차량이 가까울 때(초기 거리 및 방위 오차: 0.03 m, 3°)는



[Fig. 8] Parking path following error distribution with respect to the error postures (box-plot)

Preview 제어기의 경로 추종 오차 비용이 가장 적었으나, 목표 경로 점과 차량이 가깝지 않은 경우[Fig. 8]의 Preview 제어기 평균 오차 값은 35.198로 가장 높았고, 그 다음으로 Kanayama (31.502)와 MPC (18.371) 순서이다.

초기 오차가 있는 상황에서 서로 다른 세 가지의 곡률 값을 가진 경로를 추종한 성능도 확인하였다[Fig. 9]. 곡률은 (1/9 m)  $\cong$  0.11, (1/11 m)  $\cong$  0.09, (1/13 m)  $\cong$  0.07, 이며 초기 오차는 각각 (0.60 m, 8.5°), (0.59 m, 7.3°), (0.57 m, 6.0°)이다. 평균 거리 오차와 평균 방위 오차는 (Kanayama), (Preview), (MPC) 순서로 0.11 곡률일 때에는 (0.09 m, 2.5°), (0.11 m, 3.1°), (0.08 m, 2.6°)이고, 0.09 곡률일 때에는 (0.06 m, 2.3°), (0.09 m, 2.6°), (0.05 m, 2.2°), 0.07 곡률일 때에는 (0.06 m, 2.1°), (0.07 m, 2.7°),



[Fig. 9] Parking path following results with the three curvatures



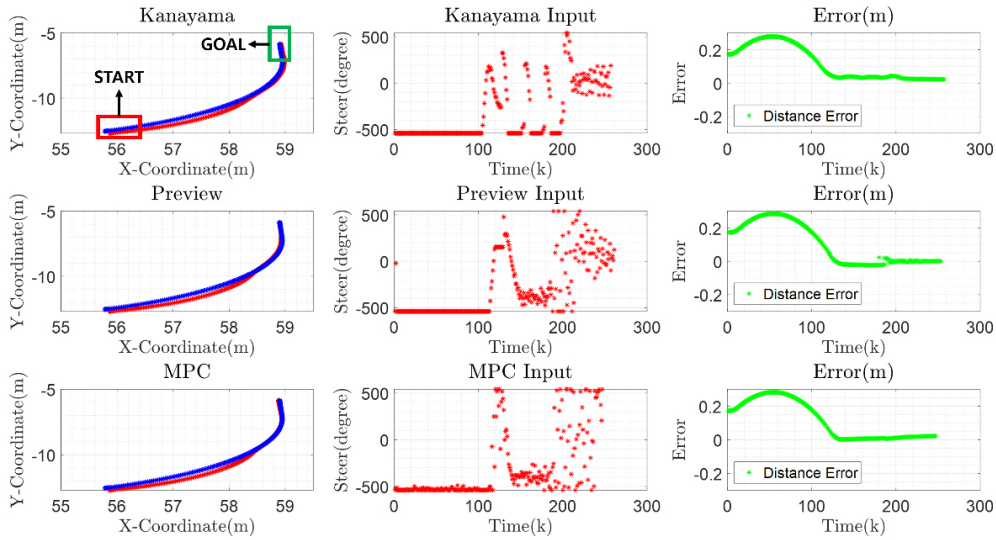
(0.05 m, 2.1°)이다. 이 결과는 4.1절의 몬테카를로 시뮬레이션을 통해 얻은 결과와 같은 경향을 지니며, 곡률이 커질수록 경로 추종 오차가 늘어남을 확인할 수 있었다.

### 4.3 제어 입력에 외란이 있을 때의 세 가지 경로 추종 알고리즘 성능 비교 분석 실험

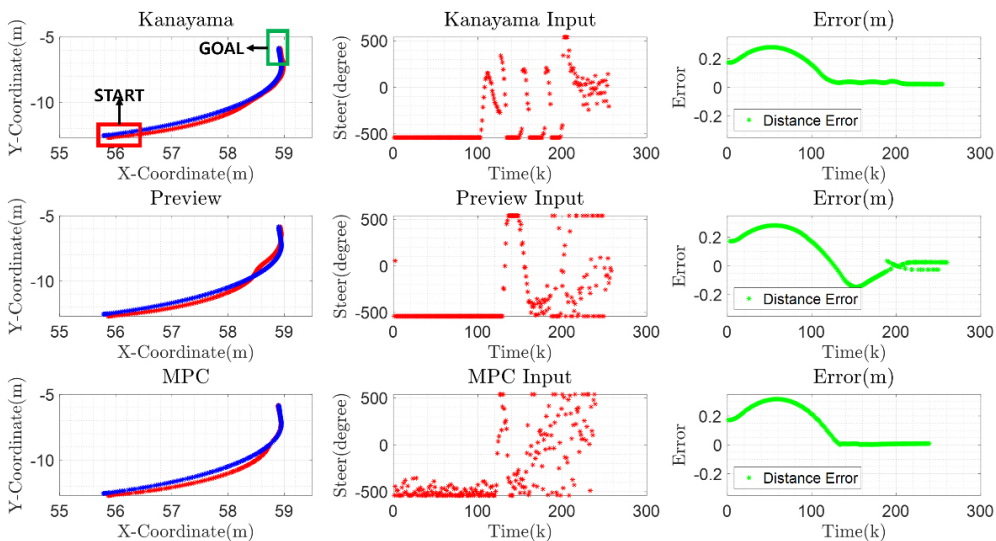
본 절에서는 차량에서 실제 조향 제어 입력이 인가되는 부분인 액추에이터에 노이즈가 있을 때의 세 가지 경로 추종 알고리즘의 성능을 평가하기 위한 시뮬레이션을 진행한다. 이 실험은 기존의 후방 목표 경로를 추종하면서, 제어 성능을 확인하기 위해 목표 점 기준([Fig. 7] 오른쪽 그림에서의 빨간

점), x축으로 0.258 m, y축으로 0.067 m, 방위로 -3.43° 더한 초기 오차 자세에서 후방 경로 추종을 하는 것이다.

제어 입력에 외란을 고려한 시뮬레이션은 기존 제어 입력에 외란을 인가하는 형태( $\delta_{input} = \delta_{original} + \delta_{disturbance}$ )로 수행됐다.  $\delta_{original}$ 은 각 제어 방법에서 구한 제어 입력이다. 여기서  $\delta_{disturbance}$  값은 평균 0이고, 표준 편차  $\left(\frac{\delta_{max}}{4}\right)^2$  또는  $\left(\frac{\delta_{max}}{2}\right)^2$  값을 이용했고 가우시안 분포(Gaussian Distribution) 형태를 가진다. 두 개의 가우시안 분포를 띄는 제어 입력 외란을 고려한 후방 경로 추종 시뮬레이션 결과는 각각 [Fig. 10]와 [Fig. 11]이다. 두 그림에서의 각 행은 세 가지 경로 추종 방법이고, 첫 번째 열은 후방 경로를 추종한 궤적, 두 번째 열은 제어 입력 값이며, 세 번째 열은 거리 오차 값이다. 제어 입력 값은 입력



[Fig. 10] Path following results with input disturbances ( $\delta_{disturbance} \sim N(0, (\frac{\delta_{max}}{4})^2)$ )



[Fig. 11] Path following results with input disturbances ( $\delta_{disturbance} \sim N(0, (\frac{\delta_{max}}{2})^2)$ )

외란 값이 더해지기 때문에, 최대 조향각 값( $\delta_{max}=540^\circ$ )으로 제한하였다.

세 가지 경로 추종 방법의 추종 성능에 대한 정량적인 결과로써 평균 거리 및 방위 오차와 주차 경로 추종의 가장 마지막 주차 지점에서의 거리 및 방위 오차를 Table로 정리하였다. [Table 2]는 제어 입력 외란이 없을 때, [Table 3]은 제어 입력 외란 값들의 표준 편차 값이  $\left(\frac{\delta_{max}}{4}\right)^2$  일 때의 결과 값이고, [Table 4]는 제어 입력 외란 값들의 표준 편차 값이  $\left(\frac{\delta_{max}}{2}\right)^2$  일 때의 결과를 보여준다. 제어 입력 외란이 없는 실험 결과인 [Table 2]에서 보면, 평균 거리 오차는 MPC가 가장 작으며, 방위 오차는 Kanayama 방법이 가장 작다. 또한, 세 가지 방법으로 모두 주차 경로 추종 완료 지점에서의 거리 및 방위 오차를 보면, 2 cm, 0.5° 이하로 경로가 추종 됨을 확인할 수 있다. 그러나, [Table 3]에서 볼 수 있듯이 제어 입력 외란이 가해질 때, 세 가지 경로 추종 방법 모두 거리 및 방위 오차 평균 값이 증가하는 경향을 볼 수 있으며 특히, Preview 제어기의 방위 오차 값이 가장 많이 증가된 것을 확인할 수 있다. 하지만, 주차 목표점에서의 거리 및 방위 오차는 세 가지 방법 모두 2 cm 내외의 거리 오차와 1°이하의 방위 오차를 보이기 때문에 주차 경로 추종이 적절하게 이루어졌음을 확인할 수 있다.

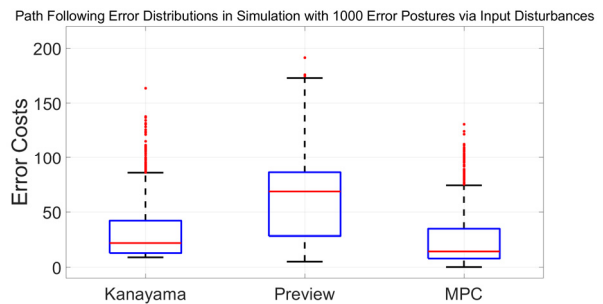
[Table 4]는 제어 입력 외란을 더 크게 입력한 것으로 외란 값들의 표준 편차 값이  $\left(\frac{\delta_{max}}{2}\right)^2$  일 때의 경로 추종 결과의 의미하며 세 가지 경로 추종 방법의 오차 평균값은 증가하였다. Preview 제어기는 제어 입력 외란이 없을 때를 기준으로 하여 가장 큰 비율로 오차 평균값이 증가하였다(거리 오차: 15.7%, 방위 오차: 34.6%). 반면에, Kanayama 제어기는 세 가지 방법

중에서 가장 낮은 비율(거리 오차: 0.9%, 방위 오차 1.8%)로 오차 값이 증가했기 때문에, 세 가지 방법 중에서 제어 입력 외란에 가장 강인함을 보였다.

4.4 초기 경로 추종 오차와 제어 입력에 외란이 있을 때, 세 가지 경로 추종 알고리즘 성능 비교 분석 실험

본 절에서는 4.1 절에서 몬테카를로 시뮬레이션을 통해 구한 최적 제어 입력 파라미터를 이용하고 4.2 절에서 정의한 1,000가지의 초기 오차 자세에서, 4.3 절에서 언급된 제어 입력 외란이 존재하는 시나리오에서 경로 추종 성능을 확인하고자 한다. 여기서 이용한 제어 입력 외란은 평균이 0이고, 표준 편차 값이  $\left(\frac{\delta_{max}}{2}\right)^2$  인, 가우시안 분포를 이용하였다.

세 가지 경로 추종 방법을 이용한 결과에 대한 box 그래프는 [Fig. 12]에서 확인할 수 있다. 제어 입력 외란이 없는 [Fig. 8]과 비교하면 Preview 제어기와 MPC 제어기의 오차 분포가



[Fig. 12] Parking path following error distribution with error postures and input disturbances(box-plot)

[Table 2] Path following errors of a parking path without the input disturbance

	Avg. Distance Error (m)	Avg. Orientation Error (°)	Final Distance Error (m)	Final Orientation Error (°)
Kanayama	0.109	2.554	0.021	0.442
Preview	0.102	2.883	0.008	0.331
MPC	0.094	2.834	0.006	0.239

[Table 3] Path following errors of a parking path with the input disturbance ( $\delta_{disturbance} \sim \mathcal{N}(0, (\frac{\delta_{max}}{4})^2)$ )

	Avg. Distance Error (m)	Avg. Orientation Error (°)	Final Distance Error (m)	Final Orientation Error (°)
Kanayama	0.110	2.581	0.022	0.273
Preview	0.108	3.351	0.001	0.885
MPC	0.113	2.972	0.021	0.462

[Table 4] Path following errors of a parking path with the input disturbance ( $\delta_{disturbance} \sim \mathcal{N}(0, (\frac{\delta_{max}}{2})^2)$ )

	Avg. Distance Error (m)	Avg. Orientation Error (°)	Final Distance Error (m)	Final Orientation Error (°)
Kanayama	0.110	2.602	0.022	0.310
Preview	0.121	4.411	0.023	1.053
MPC	0.129	3.172	0.009	0.199

[Table 5] Path following results with error postures and the input disturbance ( $\delta_{disturbance} \sim N(0, (\frac{\delta_{max}}{2})^2)$ )

	W/O Input Disturbances		W/ Input Disturbances		Rate of Increase	
	Error Cost	Control Cost	Error Cost	Control Cost	Error Cost (%)	Control Cost (%)
Kanayama	31.50	38.3	31.78	49.7	0.89	29.76
Preview	35.19	31.77	61.27	78.84	74.16	148.16
MPC	18.37	92.39	25.93	121.98	41.15	32.03

더 넓어졌음을 확인할 수 있다. 이에 반해, Kanayama 제어기는 제어 입력 외란이 있을 때와 없을 때와의 큰 차이가 없었다. Preview 제어기는 오차 중간값이 크게 올랐으며, MPC 제어기도 오차의 중간 값과 최대 값이 각각 증가되었음을 확인할 수 있다.

제어 입력 외란의 영향에 대한 오차 비용 증가율은 [Table 5]에 정리하였다. 제어 입력 외란이 없을 때, MPC가 경로 추종 오차 비용이 가장 낮다. 하지만, 제어 노력에 해당하는 Control Cost 비용이 큰 것으로 보아, 제어 입력 변화율이 큰 것을 확인할 수 있었다. 제어 입력 외란이 있을 때, Preview 제어 방법이 가장 큰 오차 비용을 보였고 제어 입력 외란이 없을 때 비해서 가장 큰 비용 증가율을 보였다. 제어 입력 외란이 있어도 MPC의 제어 방법은 가장 적은 오차 비용을 보였으나 외란이 없을 때와 마찬가지로 가장 큰 제어 노력 비용을 보였다. 이러한 부분은 MPC 제어 방법에서 제어 입력 변화율에 대한 제약 조건을 이용하면 제어 노력 비용의 감소를 기대할 수 있으나 제어 입력의 변화율 제약은 곧 경로 추종 성능에도 영향을 미칠 수 있는 상충(Trade-off)관계가 있기 때문에 제어기 설계자는 이러한 부분을 고려해야 한다. 제어 입력 외란에 대해서 세 가지 방법 중 가장 강인한 제어 방법은 Kanayama 제어기였으며, 제어 오차 비용과 제어 노력 비용에서 가장 변화가 적었다.

## 5. 결론

본 연구는 기존에 경로 추종 방법으로 제안되었던 알고리즘들의 특징에 대해 설명을 하고 자율 주차에 적용될 수 있는 세 가지 경로 추종 알고리즘(Kanayama, Preview 및 MPC)을 소개하였다. 주차 경로와 같이 곡률이 크고, 경로 길이가 짧아서 경로 끝 점이 존재하는 주행 상황에서, 세 가지 경로 추종 알고리즘의 정성적/정량적 비교 분석이 수행되었다. 우선, 세 가지 경로 추종 방법의 최적 제어 파라미터를 결정하기 위해 몬테카를로 시뮬레이션이 수행되었다. 그리고 자율 주차 주행 중 차량 방향 전환 지점에서의 초기 오차에 대해 각 제어기 별 성능이 확인되었다. 또한, 차량의 핸들을 제어할 때, 조향 액추에이터에서 발생할 수 있는 노이즈를 제어 입력 외란으로 정의하고, 외란 크기 별 주차 경로 추종 성능도 확인하였다. 마지막으로, 초기 오차와 제어 입력 외란이 동시에 존재하는 환경

에서 경로 추종 성능을 비교 분석하였다. 세 가지 경로 추종 방법의 시뮬레이션 결과를 보면, 차이가 크지 않기 때문에 모두 주차 제어에 사용 가능하지만, 몇 가지 특징들이 있고 이에 대한 각 경로 추종 알고리즘들의 장/단점을 다음과 같이 언급하고자 한다.

Kanayama 제어 방법은 추종 오차를 보상하는 Feed-back 부분과 목표 경로 점들의 곡률을 이용한 Feed-forward 부분으로 제어 입력이 구성 되어 있다. 이 방법은 세 가지 경로 추종 알고리즘 중에, 가장 큰 경로 추종 오차를 보이지만, 세 알고리즘 중에서 구현이 간단하고, 주차 목표점에서도 주차 경로 추종의 성능이 각각 4 cm, 1° 내외의 거리 및 방위 오차를 보이기 때문에 주차 경로 추종 알고리즘으로 적합함을 확인하였다. 특히, 이 방법은 제어 입력 외란이 존재 할 때에도 세 가지 알고리즘 중 가장 강인한 경로 추종 성능을 보였으며, 세 가지 경로 추종 방법 중에서 제어 노력 비용이 가장 적기 때문에 효율적으로 주차 경로 추종을 수행할 수 있다.

Preview 제어기는 일정한 예견 거리에서의 목표 점과 현재 차량에서 가장 가까운 목표 점에서의 경로 추종 오차를 줄일 수 있는 제어 방법으로써 몬테카를로 시뮬레이션을 통해 나온 경로 추종 성능은 세 알고리즘 중에서 가장 나았다. 그러나, 이 방법은 차량의 방향이 전환되는 위치에서 오차가 발생했을 때 세 알고리즘 중 제어 성능이 가장 큰 비율로 감소 되었으며 제어 입력 외란이 있을 때, 경로 추종 비용의 변화율이 가장 높아서 제어 입력 외란에 취약함을 보였다.

MPC 제어기는 미리 정의해 놓은 비용을 최소화하면서 동시에 제약 조건을 고려하여 최적 제어 입력을 구하는 방식이다. 세 가지 알고리즘 중 MPC 제어기는 초기 오차가 있을 때와 제어 입력 외란이 있을 때 모두 가장 적은 추종 오차 비용을 보였고, 주차 목표점에서도 거리 오차 및 방위 오차가 가장 적었다. 그러나, 세 가지 제어기 중, Kanayama 제어기는 제어 입력 외란에 가장 강인한 반면, MPC는 제어 입력 외란이 있을 때 추종 오차 비용의 증가율이 더 높았다. 또한, MPC는 제어 노력 비용에서 세 제어기 중, 가장 많은 비용을 사용 하였으며 이는 차량의 조향 액추에이터에 상대적으로 큰 부하가 걸릴 수 있고, 조향의 변화는 곧 차량의 방위의 변화를 유발한다. 이러한 변화는 차량의 횡 방향 가속도를 생성할 수 있기 때문에 결론

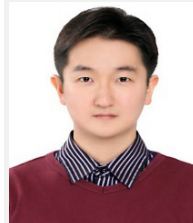
적으로, 승차감에도 악영향을 미칠 수 있다. MPC 제어기는 초기 경로 추종 오차와 제어 입력 외란이 있음에도 불구하고, 주차 목표점에서 2 cm, 1°이하의 경로 추종 오차를 보여 주차 경로 추종 알고리즘으로 적합함을 보였다.

본 연구의 후속 연구로써 위치 추정 오차가 있는 실제 차량에서도 주차 경로 추종 성능을 확인 하고자 한다.

## References

- [1] M. Kim and J. Park, "Autonomous Parking using Sampling-Based Path Planning and Model Predictive Control," *The Korean Society of Automotive Engineers Annual Spring Conference*, pp. 879-901, 2018, [Online], <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07546720>.
- [2] M. Kim, G. Im, J. Ahn, M. Kim, and J. Park, "Autonomous parking algorithm using automatic selection of forward and backward switching points via model predictive control," *The Korean Society of Automotive Engineers Annual Autumn Conference*, pp. 646-649, 2018, [Online], <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07593207>.
- [3] S. Lee, M. Kim, Y. Youm, and W. Chung, "Control of a car-like mobile robot for parking problem," *1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, Detroit, MI, USA, 1999, DOI: 10.1109/ROBOT.1999.769891.
- [4] R. W. Roger, "Asymptotic stability and feedback stabilization," *Differential Geometric Control Theory*, pp. 181-191, 1983, [Online], <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.324.9912>.
- [5] M. Kim, S. Shin, and J. Park, "Study on vehicle lateral control for backward driving," *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Xi'an, China, 2016, DOI: 10.1109/URAI.2016.7625734.
- [6] I. E. PAROMTCHIK and C. LAUGIER, "Autonomous parallel parking of a nonholonomic vehicle," *Conference on Intelligent Vehicles*, Tokyo, Japan, pp. 13-18, 1996, DOI: 10.1109/IVS.1996.566343.
- [7] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Staven, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, K. Lau, C. Oakly, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummei, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661-692, 2006, DOI: 10.1002/rob.20147.
- [8] R. Wallace, A. Stentz, C. Thorpe, H. Moravec, W. L. Whittaker, and T. Kanade, "First Results in Robot Road-Following," *International Joint Conference on Artificial Intelligence*, 1985, [Online], <https://www.semanticscholar.org/paper/First-Results-in-Robot-Road-Following-Wallace-Stentz/aed962d06b081820cb3481fafa5a59568fca4764>.
- [9] *The 2005 DARPA grand challenge: the great robot race*, M. Buehler, K. Iagnemma, and S. Singh eds., Springer Nature, Switzerland, 2007, [Online], <https://www.springer.com/gp/book/9783540734284>.
- [10] R. Rajamani, C. Zhu, and L. Alexander, "Lateral control of a backward driven front-steering vehicle," *Control Engineering Practice*, vol. 11, no. 5, pp. 531-540, 2003, DOI: 10.1016/S0967-0661(02)00143-0.
- [11] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Nogu, "A stable tracking control method for an autonomous mobile robot," *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA, pp. 384-389, 1990, DOI: 10.1109/ROBOT.1990.126006.
- [12] W. Yang, L. Zheng, Y. Li, Y. Ren, and Y. Li, "A Trajectory Planning and Fuzzy Control for Autonomous Intelligent Parking System," *WCX<sup>TM</sup> 17: SAE World Congress Experience*, 2017, DOI: 10.4271/2017-01-0032.
- [13] J. Keighobadi and Y. Mohamadi, "Fuzzy Sliding Mode Control of a Non-Holonomic Wheeled Mobile Robot," *2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, Smolenice, Slovakia, 2011, DOI: 10.1109/SAMI.2011.5738888.
- [14] Z. Fan and H. Chen, "Study on Path Following Control Method for Automatic Parking System Based on LQR," *Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 10, no. 1, pp. 41-49, 2017, DOI: 10.4271/2016-01-1881.
- [15] M. Brezak, P. Ivan, and P. Nedjeljko, "Experimental comparison of trajectory tracking algorithms for nonholonomic mobile robots," *2009 35th Annual Conference of IEEE Industrial Electronics*, Porto, Portugal, 2009, DOI: 10.1109/IECON.2009.5415188.
- [16] D. HeB, M. Althoff, and T. Sattel, "Comparison of trajectory tracking controllers for emergency situations," *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast, QLD, Australia, 2013, DOI: 10.1109/IVS.2013.6629465.
- [17] D. Calzolari, B. Schürmann, and M. Althoff, "Comparison of trajectory tracking controllers for autonomous vehicles," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, 2017, DOI: 10.1109/ITSC.2017.8317800.
- [18] S. Dominguez, A. Ali, G. Garcia, and P. Martinet, "Comparison of lateral controllers for autonomous vehicle: Experimental results," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016, DOI: 10.1109/ITSC.2016.7795743.
- [19] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, 2012, DOI: 10.1007/s11081-011-9176-9.
- [20] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, DOI: 10.1109/IROS.2013.6696520.
- [21] Kim Minsung, Github, [Online], <https://github.com/WestsideMS>, Accessed: Feb. 04, 2020.

- [22] M. Mark, I. A. Şucan, and L. E. Kavraki, "Benchmarking Motion Planning Algorithms: An Extensible Infrastructure for Analysis and Visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96-102, 2015, DOI: 10.1109/MRA.2015.2448276.
- [23] F. A. Cheein and G. Scaglia, "Trajectory tracking controller design for unmanned vehicles: A new methodology," *Journal of Field Robotics*, vol. 31, no. 6, pp. 861-887, 2014, DOI: 10.1002/rob.21492.



**김민성**

2010 중앙대학교(공학사)  
2011~현재 서울대학교 융합과학기술대학원 석사, 박사 통합과정

관심분야: Autonomous Vehicle, Motion Control, Optimal Control



**임규범**

2018 국민대학교(공학사)  
2018~현재 서울대학교 융합과학기술대학원 석사 과정

관심분야: SLAM, Autonomous Vehicle



**박재흥**

1995 서울대학교 항공우주공학과(공학사)  
1999 서울대학교 항공우주공학과(공학석사)  
2006 Stanford University Aero/Astro(공학박사)  
2009~현재 서울대학교 융합과학기술대학원 교수

관심분야: Robot-environment Interaction, Contact Force Control, Multi-Contact Control, Whole-body Dynamic Control, Bio-mechanics